CAR-TR-703
CS-TR-3222

F49620-93-1-0039
DACA76-92-C-0009
February 1994

RECOGNITION BY FUNCTIONAL PARTS

Ehud Rivlin
Department of Computer Science
Technion
Israel Institute of Technology
Haifa, Israel

Sven J. Dickinson
Department of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 1A4

Azriel Rosenfeld
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

We present an approach to function-based object recognition that reasons about the functionality of an object's intuitive parts. We extend the popular "recognition by parts" shape recognition framework to support "recognition by functional parts", by combining a set of functional primitives and their relations with a set of abstract volumetric shape primitives and their relations. Previous approaches have relied on more global object features, often ignoring the problem of object segmentation and thereby restricting themselves to range images of unoccluded scenes. We show how these shape primitives and relations can be easily recovered from superquadric ellipsoids which, in turn, can be recovered from either range or intensity images of occluded scenes. Furthermore, the proposed framework supports both unexpected (bottom-up) object recognition and expected (top-down) object recognition. We demonstrate the approach on a simple domain by recognizing a restricted class of hand-tools from 2-D images.

Keywords: function-based object recognition, recognition by parts, abstract shape reasoning

The support of the Air Force Office of Scientific Research, under Grant F49620-93-1-0039, and the Advanced Research Projects Agency (ARPA Order No. 8459) and the U.S. Army Topographic Engineering Center under Contract DACA76-92-C-0009 is gratefully acknowledged, as is the help of Sandy German in preparing this paper.

1 Introduction

The problem of object recognition from sensory data is defined in the literature as the association of visual input with a name or symbol. In the absence of distinguishing properties such as color, texture, or motion, object recognition first requires the visual recovery of shape, followed by the matching of the recovered shape to a database of known objects [16]. Although much research on the topic has been published, the community still lacks vision systems that can recognize in real time a large number of objects (natural or man-made). Full recovery has been difficult to achieve while matching suffers from combinatorial explosion.

Model-based recognition has been suggested as a remedy to these problems. Many 3-D object recognition systems take a single object model and attempt to locate it in the image, e.g., [13, 14, 34]. The object models are commonly CAD-like, capturing the *exact* geometry of the object. Although very effective for certain robot vision tasks in constrained environments, where a known target must be accurately localized for manipulation or inspection, these techniques are inadequate when addressing less constrained environments.

Consider, for example, a robot vision system whose goal is to move through a handicapped person's household, retrieving and manipulating everyday objects such as books, cups, chairs, etc. How can we avoid having to provide the system with detailed CAD specifications of each object that the system is to recognize? One way of making object models more flexible is to parameterize geometric models, as proposed by Brooks in his ACRONYM system [3]. For example, the legs of a chair model could have lengths that fall in some specified range, or the number of chair legs could be variable. Object recognition systems using parameterized models have also been proposed by Huttenlocher [12] and by Lowe [15]. However, all three of the above systems are very top-down, requiring not only knowledge of what object is in the image, but in some cases a good initial guess as to the orientation of the object.

A different approach to the problem is to consider the recognition process in the context of an agent interacting with its environment [23]. The recognition process is subordinate to the agent's intentions and behavior in its environment. Recognition is equivalent to the process that checks if an object suits a particular purpose. If an object is perceived to fulfill a function necessary to carry out a certain behavior or action, then it is recognized. Gibson's theory of affordances [10], i.e., properties that are defined with reference to an observer, was a major step in this direction. Winston et al. [35] emphasized how much easier it is to describe what objects are used for, rather than to describe what objects look like. They tried to show how recognition could be performed using functional definitions, and how physical models could be learned using functional definitions and specific acts of identification.

When we consider recognition from a functional point of view, we replace the concept of shape-alone based recognition by a more general and flexible approach. For example, if we wish to model four chairs, each having a different configuration of differently shaped parts but all functioning as chairs, we would require four different object shape models. Alternatively, recognition based on functionality would enable our mobile robot to possess knowledge of the needed function of a chair without explicitly specifying the possible shape of a chair. The seminal work of Stark and Bowyer et al. [25]–[31] has addressed function-based object recognition, focusing on domains that include such objects as chairs and dishes. They define a set of functional primitives specific to each object class. For example, in their system that recognizes chairs, they have functional primitives for support, sitting height, stability, etc. From a CAD representation of an object, they can compute these primitives and categorize the object. Although their system has been tested mainly with CAD data, they have also applied the system to range images acquired by an Odetics range scanner.

Despite the success of their approach, it has some limitations. To begin with, the approach assumes a 3-D representation of the image from which they can compute the functional primitives. Furthermore, the approach assumes an image of an isolated object; object occlusion in the image cannot be supported since no object segmentation is performed on the image data. Thus the work of Stark and Bowyer takes a *global* approach to functional recognition, making it sensitive to occlusion and partial views. Their functional reasoning does not extend to function-based recognition from 2-D imagery containing multiple occluded objects.

In this paper, we present a theory of function-based recognition which is a natural extension of part-based shape recognition. Instead of focusing on global properties such as stability, height, existence of large horizontal surfaces, etc., we will reason about the functionality of an object's parts. Moreover, those parts are the *same* parts that we recover from the image for shape recognition. Thus, instead of reasoning about the functionality of a collection of 3-D points or planar surfaces, we propose to reason about a more intuitive notion of an object's parts (Pentland [19]). Although we will not index using part shape, we can use knowledge of part shape to help segment the image into parts. Given a set of recovered volumetric parts, we can then reason about both the functionality of the individual parts and the interactions between the parts. Such interactions can include relative orientation, size, shape, or even motion!

Returning to our mobile robot example, when the robot encounters an object, it first segments the object into a set of volumetric parts, supporting part recovery from incomplete views of the object and supporting object occlusion. Once the robot decides which parts belong to the object, it can then infer the possible functionalities of individual parts and collections of parts. The robot can check if the functionality needed for a certain action is consistent with the recovered functionality. In contrast with the approach of Stark and Bowyer, when searching the image for a "chair kind of support", we would like to reason about a set of chair legs, a seat, and a back, rather than a set of simple planar surfaces or 3-D points.

In the following sections, we outline our theory of object functionality in Section 2, and introduce a representation for volumetric parts from which we reason about functionality. Section 3 discusses the recovery of the volumetric parts from both 3-D range and 2-D intensity images. In Section 4 we describe our recognition algorithm as it applies to both expected (top-down) and unexpected (bottom-up) object recognition. Finally, in Section 5, we demonstrate how the technique is applied to the domain of hand tools.

2 Representing Object Functionality

Our theory of function-based object recognition is a natural extension of part-based shape recognition. That is, we reason about the functionality of an object's parts and their interrelations. Figure 1 illustrates the concept. At the shape level, objects are constructions of coarse volumetric primitives with spatial relations between the primitives. At the function level, the shape primitives map to a set of functional primitives and the spatial relations map to a set of functional relations. At the functional level, objects are not represented in terms of shape, but in terms of a set of functional primitives and relations. In the following sections, we describe this hierarchical representation in more detail. We begin by describing the coarse shape representation and follow with the functional representation. Finally, we illustrate the representation by means of an example.

2.1 Representing Shape

2.1.1 Shape Primitives

Our shape representation models objects as constructions of coarse volumetric shapes belonging to four classes: sticks, strips, plates, and blobs. The representation is an extension to the generalized blob models (sticks, plates, and blobs) proposed by Mulgaonkar, Shapiro, and Haralick [18]. Our

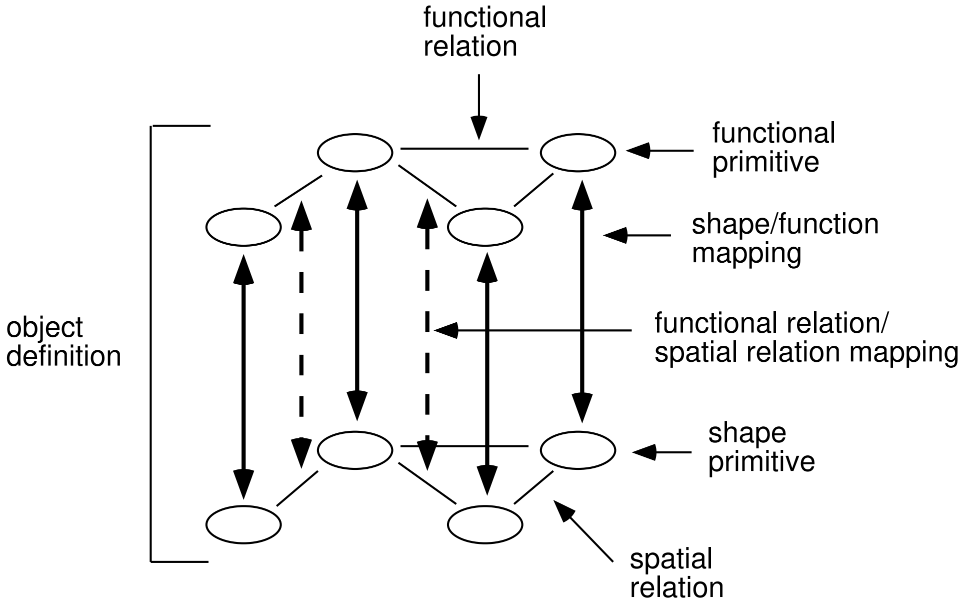


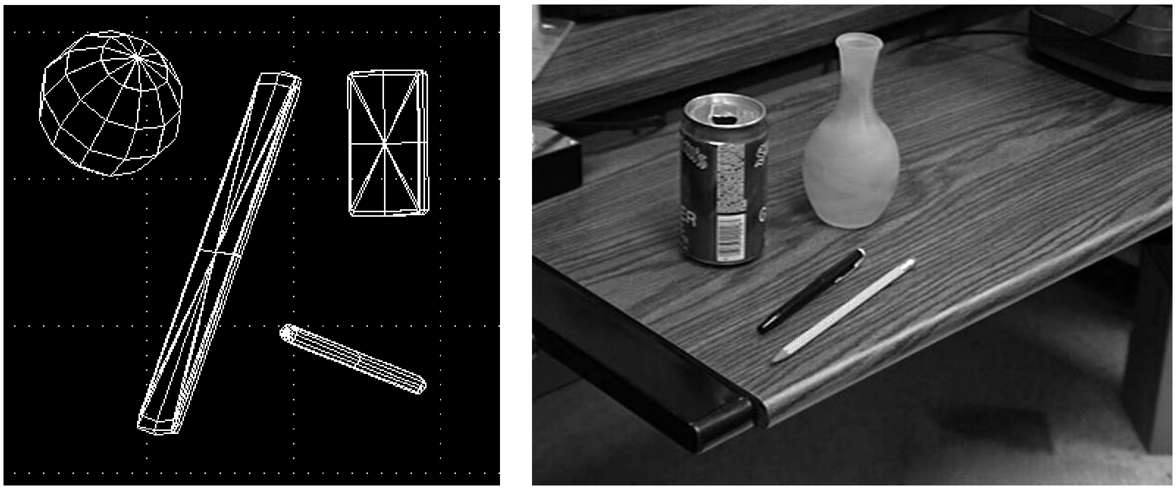
Figure 1: Representing object functionality

four classes are distinguished by their relative dimensions. Letting a_1, a_2, and a_3 represent the length, width, and height, respectively, of a volumetric part, we can define the four classes as follows:

$$\begin{aligned} \mathit{Stick}: &\quad a_1 \simeq a_2 \ll a_3 \vee a_1 \simeq a_3 \ll a_2 \vee a_2 \simeq a_3 \ll a_2 && (1)\\ \mathit{Strip}: &\quad a_1 \neq a_2 \wedge a_2 \neq a_3 \wedge a_1 \neq a_3 && (2)\\ \mathit{Plate}: &\quad a_1 \simeq a_2 \gg a_3 \vee a_1 \simeq a_3 \gg a_2 \vee a_2 \simeq a_3 \gg a_2 && (3)\\ \mathit{Blob}: &\quad a_1 \simeq a_2 \simeq a_3 && (4) \end{aligned}$$

Intuitively, if all three dimensions are about the same, we have a blob. If two are about the same and the third is very different, we have two cases: if the two are bigger than the third, we have a plate, while if the two are smaller than the third, we have a stick. Finally, when no two dimensions are about the same, we have a strip. For example, a knife blade is a strip, because no two of its dimensions are similar.

Figure 2 shows examples of these four types of primitives. The ratios of the three dimensions for the stick are 1:1:10; for the strip, 48:1:7; for the plate, 16:16:1; and for the blob, 1:1:1. Note that our taxonomy of primitives is qualitative; a cylinder, for example, can be a stick or a plate. Variations of the basic primitives, such as tapering, pinch, curvature, and so on, can also be considered in the context of functionality.



(a) (b)

Figure 2: Shape primitives. (a) The four primitives modeled by superquadrics using Pentland's Thingworld [19]. (b) Two blobs (can and vase), two sticks (pencil and pen), and a plate (table).

2.1.2 Spatial Relations

We can qualitatively describe the ways in which two shape primitives can be combined. For example, we can attach two shapes end-to-end, end-to-side, or side-to-side, as proposed by Biederman when building objects out of geons [2]. To further specify these attachments, we adopt the convention of labeling each volumetric primitive's attachment surfaces [7]. For example, a square plate has six attachment surfaces, while a cylindrical stick has three attachment surfaces. For simplicity, we shall require any junction of two primitives to involve exactly one attachment surface from each primitive. Figure 3 illustrates a simple object composed of a blob and a stick; note that in this example, both objects are cylinders.

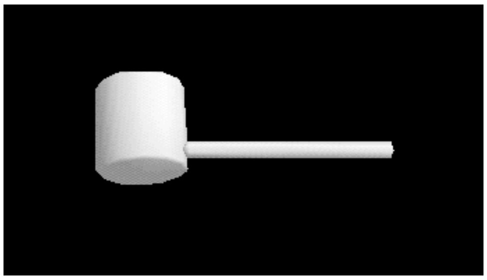


Figure 3: Concatenation: a blob and a stick.

In addition to specifying the two attachment surfaces that participate in the junction of two primitives, we can also consider the angle at which they join, and we can classify the joints as perpendicular, oblique, tangential, etc. Another refinement would be to qualitatively describe the position of the joint on each surface. For example, an attachment can be near the middle, near the side, near the corner, or near the end of a surface. Figure 4 illustrates six different concatenations of a strip and a stick. A stick has three attachment surfaces, but when symmetry is taken into account it has only two distinguishable attachment surfaces. Similarly, a strip has three distinguishable attachment surfaces. Therefore, there are six ways of combining the two primitives in terms of attachment surfaces if we ignore where on a surface an attachment is made as well as the angle of the attachment.

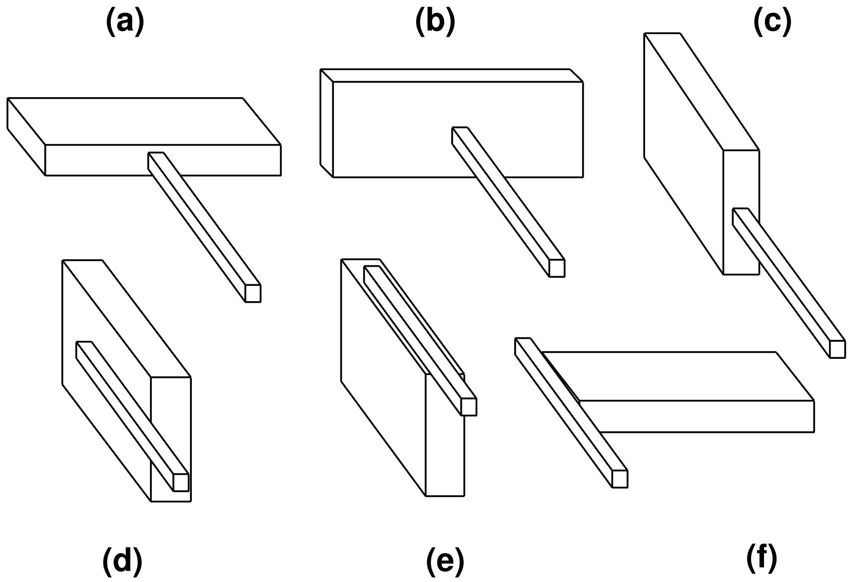


Figure 4: Six different objects composed of a strip and a stick.

2.2 Representing Function

Functional Primitives

Functional primitives represent the building blocks of a functional representation of an object. For example, the functional primitives defining a coffee cup would include a handle and a container; a chair would include a seat, a base, and a back [25, 26]. In the remainder of this paper, we will illustrate our approach to functional object recognition by focusing on a class of manipulation tasks. Bearing in mind that a manipulation task involves an agent and its environment, we will define a class of objects that have an *end-effector* (that part which delivers the action) and a *handle* (that part which provides the interface between the agent and the end-effector). Examples of such objects might include simple hand tools like a screw driver or a hammer, or everyday objects like cups, glasses, or plates.

2.3 Mapping Shape to Function

In general, the mapping between shape primitives (and their relations) and functional primitives is many-to-one. For example, three or more chair legs may satisfy the functional primitive of chair base. For simplicity, we will restrict ourselves to object models with a one-to-one mapping between shape primitives and functional primitives. Consider, for example, the functional model for a hammer specifying an end-effector and a handle. The end-effector should be blob-like, ensuring that the dimensions of the striking surface are roughly equal (rotationally symmetric to allow striking error in any direction). If the end-effector were stick-like, the distance between the handle junction and the striking surface would be large, making it more difficult to locate the nail. If the end-effector were plate-like, it would have insufficient momentum for driving a nail. The handle, on the other hand, should be stick-like, small enough so that it can be grasped by a human hand, and long enough to provide a high moment at its junction with the end-effector.

Functional Relations

Several parts might independently satisfy the requirements for an end-effector or a handle. However, they must be joined in a particular way in order to satisfy the requirements of a particular task. The set of functional relations linking the functional primitives describes the function of the interaction between the functional primitives. In the hammer example, the functional relation linking the handle and end-effector specifies that the handle is used to swing the end-effector in a direction which maximizes the force tangential to the swing arc while maximizing striking stability.

2.4 Mapping Function Relations to Spatial Relations

The specification of how the functional components defining an object are combined is captured by a set of functional relations. These functional relations are then mapped to a set of spatial relations linking the shape primitives. In the hammer example, the functional relation maps to an attachment between the stick (handle) and the blob (end-effector) such that the axis of the stick is orthogonal to the (principal) axis of the blob and is attached to the centroid of the blob. The complete model for the hammer, including functional and shape primitives, functional and shape relations, and the mapping between functional shapes and relations to spatial shapes and relations is outlined in Figure 5.

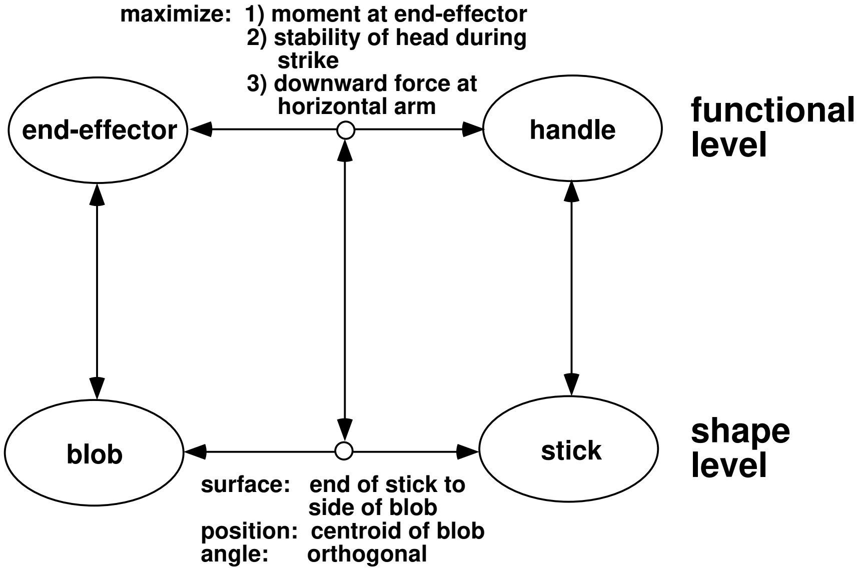


Figure 5: Functional model for a hammer.

3 Recovering Shape

In the last section, we described a set of functional primitives defined on a set of shapes consisting of sticks, strips, plates, and blobs. Since these four shape classes are defined on the basis of their relative dimensions, we need to not only segment an input image into parts, but recover 3-D (dimensional) information from those parts. In this section, we describe an approach to recovering sticks, strips, plates, and blobs from an image. The approach consists of recovering a superquadric from the image, providing explicit dimensions which we can then use to classify our shape. Superquadrics offer a compact, coarse, volumetric description of an object's parts [19]. If finer shape modeling is required, deformable superquadrics can be used to capture both global part shape (using a superquadric) and local shape (using a deformable mesh) [32]. Since superquadrics capture more shape attributes than just the x, y, and z dimensions of a part, they provide us with a foundation from which to recover a richer vocabulary of qualitative shapes with which to reason about function. For example, we may decide to distinguish among curved-axis vs. straight-axis shapes or tapering vs. constant cross-sectional sweep rules [2].

Recently, several researchers have proposed segmentation techniques for partitioning image or range data, in order to automate the process of fitting superquadric volumetric primitives to data. Most of those approaches are applied to range data only [8, 9, 11, 24], while Pentland [20] describes a two-stage algorithm to fit superquadrics to image data. In the first stage, he segments the image using a filtering operation to produce a large set of potential object "parts", followed by a quadratic optimization procedure that searches among these part hypotheses to produce a maximum likelihood estimate of the image's part structure. In the second stage, he fits superquadrics to the segmented data using a least squares algorithm. Pentland's approach is only applicable in case of occluding boundary data under simple orthographic projection, as is true of earlier work of Terzopolous et al. [33], Terzopolous and Metaxas [32], and Pentland and Sclaroff [21], which address only the problem of model fitting. In a related approach, Narayan and Jain [22] recover geons from range imagery, and use superquad fitting to determine the axis of the geon.

The approach we take, due to Dickinson and Metaxas [5, 17], is to use a qualitative segmentation of the image to provide strong constraints on the deformable model fitting procedure described in [32]. The result is a technique which allows us to recover certain classes of superquadrics from image data, under orthographic, perspective, and stereo projection [17]. Furthermore, the technique supports the recovery of occluded parts, allowing us, unlike the work of Stark and Bowyer, to reason about the functionality of objects that are only partially visible. We will not describe the above recovery methods in this paper; details can be found in [5, 17]. We will, however, describe the geometry of a deformable superquadric and show how we classify a superquadric as a stick, strip, plate, or blob.

3.1 Geometry of a Deformable Superquadric

Geometrically, the models that we can recover from either range or image data are closed surfaces in space whose intrinsic (material) coordinates are $\mathbf{u} = (u, v)$, defined on a domain Ω. The positions of points on the model relative to an inertial frame of reference Φ in space are given by a vector-valued, time varying function of $\mathbf{u}$:

$$\mathbf{x}(\mathbf{u}, t) = (x_1(\mathbf{u}, t), x_2(\mathbf{u}, t), x_3(\mathbf{u}, t))^\top, \tag{5}$$

where $^\top$ is the transpose operator. We set up a noninertial, model-centered reference frame ϕ, as shown in Figure 6, and express these positions as:

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p}, \tag{6}$$

where $\mathbf{c}(t)$ is the origin of ϕ at the center of the model and the orientation of ϕ is given by the rotation matrix $\mathbf{R}(t)$. Thus, $\mathbf{p}(\mathbf{u}, t)$ denotes the canonical positions of points on the model relative to the model frame. We further express $\mathbf{p}$ as the sum of a reference shape $\mathbf{s}(\mathbf{u}, t)$ and a displacement function $\mathbf{d}(\mathbf{u}, t)$:

$$\mathbf{p} = \mathbf{s} + \mathbf{d}. \tag{7}$$

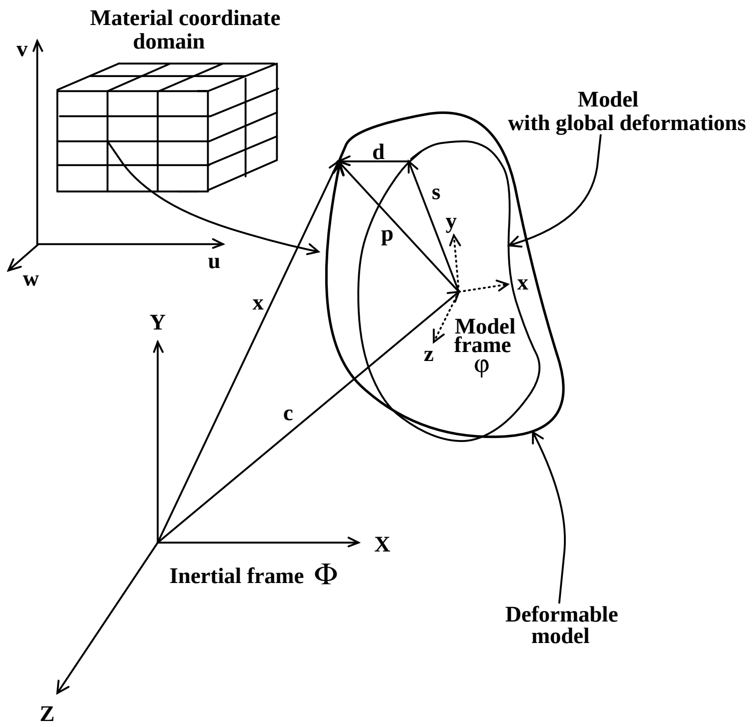


Figure 6: Geometry of a deformable model.

This formulation can be carried out for any reference shape given as a parameterized function of $\mathbf{u}$. Based on the shapes we want to recover (sticks, strips, plates, and blobs with possible tapering and bending global deformations), we first consider the case of superquadric ellipsoids [1], which are given by the following formula:

$$\mathbf{e} = a \begin{pmatrix} a_1 C_u^{\epsilon_1} C_v^{\epsilon_2} \\ a_2 C_u^{\epsilon_1} S_v^{\epsilon_2} \\ a_3 S_u^{\epsilon_1} \end{pmatrix}, \tag{8}$$

where $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v < \pi$, and where $S_w^{\epsilon} = \mathrm{sgn}(\sin w)|\sin w|^{\epsilon}$, and $C_w^{\epsilon} = \mathrm{sgn}(\cos w)|\cos w|^{\epsilon}$, respectively. Here, $a \geq 0$ is a scale parameter, $0 \leq a_1, a_2, a_3 \leq 1$ are aspect ratio parameters, and $\epsilon_1, \epsilon_2 \geq 0$ are "squareness" parameters.

We then combine linear tapering along principal axes 1 and 2, and bending along principal axis 3 of the superquadric $\mathbf{e}$[1] into a single parameterized deformation $\mathbf{T}$, and express the reference shape as

$$\mathbf{s} = \mathbf{T}(\mathbf{e}, t_1, t_2, b_1, b_2, b_3) = \begin{pmatrix} \left(\frac{t_1 e_3}{a a_3 w} + 1\right) e_1 + b_1 \cos\left(\frac{e_3 + b_2}{a a_3 w}\pi b_3\right) \\ \left(\frac{t_2 e_3}{a a_3 w} + 1\right) e_2 \\ e_3 \end{pmatrix}, \tag{9}$$

where $-1 \leq t_1, t_2 \leq 1$ are the tapering parameters in principal axes 1 and 2, respectively, and where b_1 defines the magnitude of the bending and can be positive or negative, $-1 \leq b_2 \leq 1$ defines the location on axis 3 where bending is applied and $0 < b_3 \leq 1$ defines the region of influence of bending. Our method for incorporating global deformations is not restricted to tapering and bending deformations. Any other deformation that can be expressed as a continuous parameterized function can be incorporated as our global deformation in a similar way.

We collect the parameters in $\mathbf{s}$ into the parameter vector

$$\mathbf{q}_s = (a, a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_1, t_2, b_1, b_2, b_3)^\top. \tag{10}$$

Once we have recovered a superquadric from an image (range or intensity), it is a very simple matter to extract the dimensions of the superquadric. The width (x dimension) of the superquadric is given by

$$\text{width} = a a_1, \tag{11}$$

the height (y dimension) by

$$\text{height} = a a_2, \tag{12}$$

and the length (z dimension) by

$$\text{length} = a a_3. \tag{13}$$

Given the dimensions of the part, we can classify the part as either a stick, strip, plate, or blob according to the rules described in Section 2.

3.2 An Alternative Approach to Recovering Qualitative Shape

Before we leave the recovery section, it is worth noting that there is an approach to recovering sticks, strips, plates, and blobs directly from image data without first recovering superquads. The

[1] These coincide with the model frame axes x, y and z respectively.

approach is based on the qualitative shape recovery work of Dickinson, Pentland, and Rosenfeld [7] and relies only on a region segmentation of the input image. In that approach, a fixed set of volumes was analyzed over the viewing sphere, giving rise to a set of aspects. Shape recovery was therefore formulated as part-based aspect matching with a set of conditional probabilities associated with the aspects and their features used to guide both bottom-up inferencing of features and top-down prediction of features.

This approach could be extended by analyzing the views of a greater variety of parts sorted into the four shape classes. In this manner, image dimensions of image faces in a part aspect could be measured and, using a derived conditional probability distribution, used to suggest the class membership of the part. Although much less robust than inferring class membership from a superquadric fitted to the part due to the effects of foreshortening, the approach does avoid the process of fitting the superquad.

4 Recovering Object Function

Our function-based object recognition strategy supports bottom-up (or unexpected) object recognition, whereby an object is presented to the system and the system identifies the object based on the functionalities of its parts. In addition, our strategy supports top-down (or expected) object recognition, whereby the system looks for a particular object in the image by mapping its functional parts to image feature predictions. In this section, we will describe both of these strategies.

4.1 Unexpected Object Recognition

In an unexpected object recognition task, we first first segment an input image into a set of homogeneous regions from which we recover a set of qualitative 3-D parts using local part-based aspect matching techniques [4, 6, 7]. Next, using the techniques of Dickinson and Metaxas [5, 17], we use the recovered qualitative shape to constrain the fitting of a set of deformable superquadrics to the qualitative parts. From the resulting quantitative parts, we compare the dimensions of the parts to abstract a set of sticks, strips, plates, and blobs. Furthermore, we can recover the spatial relations spanning the recovered parts.

If there is no a priori knowledge of what object is in the image, then groups of spatial primitives and their spatial relations can be used to infer a set of functional primitives and relations. The recovered functional primitives and relations are then compared to a set of functional object models.

In our simple domain of hand tools, we can map shape primitives to possible functional primitives and map shape relations to possible functional relations, providing a number of functional object hypotheses that are then compared to the object database. As an example, suppose we place a hammer in front of the camera and ask the system to identify the object. The recovery process recovers a stick and a blob in some spatial configuration. The blob maps to an end-effector as well as to all other functions a blob could serve. Similarly, the stick maps to a handle as well as to all other functions that it could serve. Finally, the spatial relation between the stick and blob maps to all functional relations joining a stick and a blob in that configuration. Combining the various interpretations for the stick, the handle, and their relationship yields a number of object hypotheses which satisfy the recovered functionality.

4.2 Expected Object Recognition

In an expected object recognition task, we use knowledge of the target object's functional model to constrain our search in the image both in terms of what we look for and where we look for it. Given a functional object model, we first choose some functional primitive whose presence in the image provides the least ambiguous mapping to the target object. For example, in looking for a cup on a table containing glasses and cups, we should look for a cup handle and not for a container since the handle is unique to the cup. Next, the functional primitive is mapped to one of the four abstract shape primitives (stick, strip, plate, and blob). Finally, the shape primitive is mapped into an image region shape prediction in terms of extent or elongation. As in the unexpected object recognition algorithm, the image is first processed to extract a region topology graph. By examining the extents (or elongations) of an image region and of its immediate neighbors, we can derive a simple heuristic for drawing attention to a particular image region. From this, we can focus the recovery of the shape primitive and constrain the search for other primitives belonging to the object.

For example, if we are searching for blobs or plates, we can rank-order the image regions by increasing extent. Regions whose immediate neighbors include a region with similar extent can be favored as being part of a blob, while regions whose neighbors do not include a region with similar extent can be favored as being part of a plate. Similarly, if we are searching for sticks or strips, we can rank-order the image regions by decreasing extent. Regions whose immediate neighbors include a region with similar extent can be favored as being part of a stick, while regions whose neighbors do not include a region with similar extent can be favored as being part of a strip. These rules can provide a useful ordering on the positions from which shape recovery is attempted.

From a candidate search position, the next step is to recover a superquadric from which 3-D part dimensions and orientation can be recovered. This is done by first recovering the qualitative shape of the part [6, 7]; this is then used to constrain the fitting of a superquadric to the image data. Once the part is verified as a stick, strip, plate, or blob, the search for other object parts can be constrained to those image regions adjacent to or in the vicinity of any previously recovered volumes.

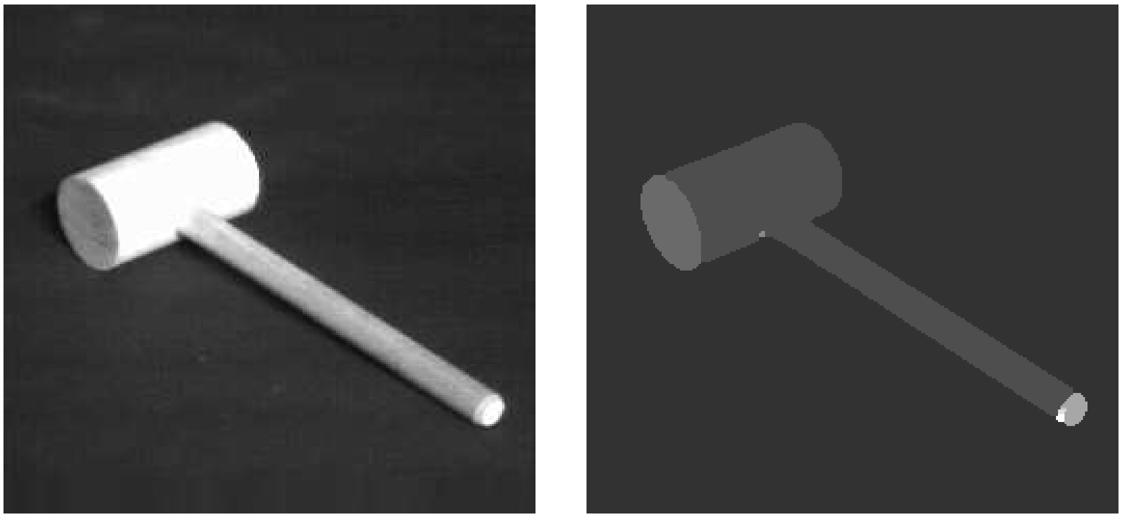
5 Results

In this section, we apply the function-based expected object recognition algorithm to the image of the mallet shown in Figure 7(a); in Figure 7(b), we show the segmented region image. Without any a priori knowledge of scene content, each of the functional primitives, namely the end-effector and handle, is deemed equally likely to appear in the image. The algorithm arbitrarily chooses the end-effector (mallet head) and maps it to a search for a blob in the image. The algorithm rank-orders regions in the image according to their ratio of area to extent (computed from the bounding box). The large region is chosen first and the bottom-up algorithm is used to recover the most likely interpretation of the region and its neighbors. The two most likely recovered volumes are shown in Figures 8(a) and (b), corresponding to the head and handle of the mallet, respectively. (Those portions of the bounding contour used to infer part identity are highlighted in the image [6, 7].)

In Figures 9(a) and (b), we show the results of using the recovered qualitative shape to constrain the fitting of a superquadric to each part; the parameters of the two superquads are given in Table 1. Since only a monocular image was used, the same arbitrary depth was chosen for both objects during the fitting stage. Without recovering the true depths of the two parts, we cannot insure that they intersect.[2] However, in this case, since the two parts intersect in the image, we assume that they intersect in 3-D.

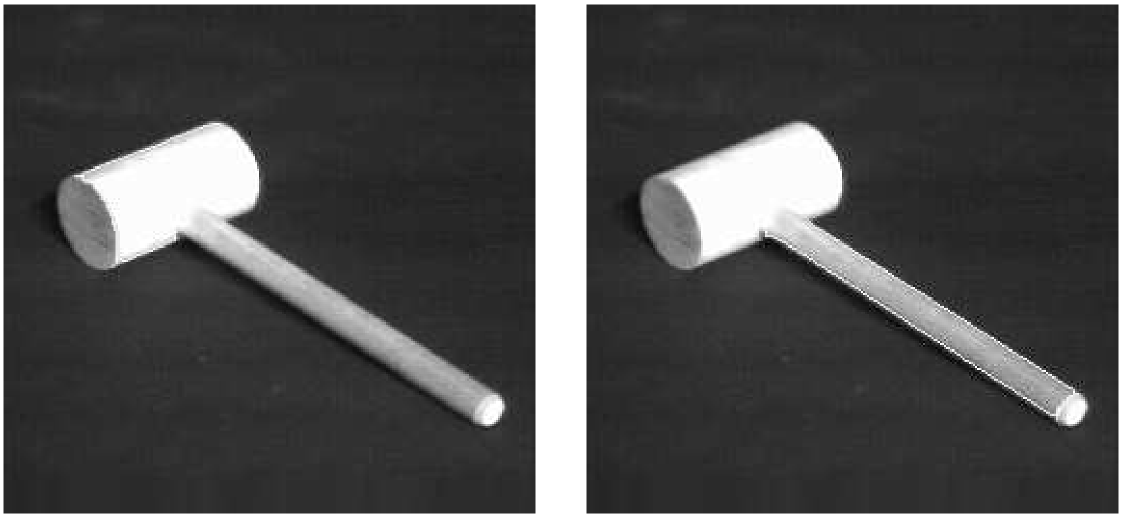
From the recovered superquad parameters in Table 1, we can proceed to classify each part as either a stick, a strip, a plate, or a blob according to equations 11, 12, and 13 in Section 2.1.1; the results are shown in Table 2. Using equations 1, 2, 3, and 4, and defining two dimensions as similar if the ratio of the bigger to the smaller is less than 4:1 (the width:height:length ratios for the two

[2]See [17] for an approach to deformable model recovery from stereo pairs.



(a) (b)

Figure 7: (a) Original image; (b) segmented region image.



(a) (b)

Figure 8: Qualitative shape recovery: (a) Most likely volume recovered for the mallet head; (b) most likely volume recovered for the mallet handle. Those contours used in inferring the recovered volume are highlighted in the image.

parts are 1:1:1.53 for the head and 1:1:5.18 for the handle), the mallet head is classified as a blob, while the mallet handle is classified as a stick.

Since our search procedure is looking for the mallet head (end-effector), it chooses the blob, and proceeds to search for the handle in the vicinity of the recovered blob. Due to region underseg-mentation, the regions corresponding to the body surfaces of the head and handle of the mallet are joined. However, those contours not used to recover the head but still belonging to the large region

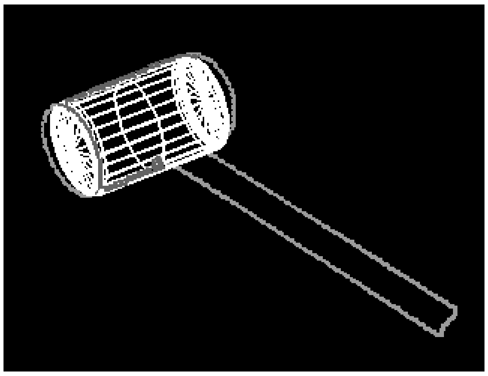
Table 1: Recovered superquad parameters for the mallet.

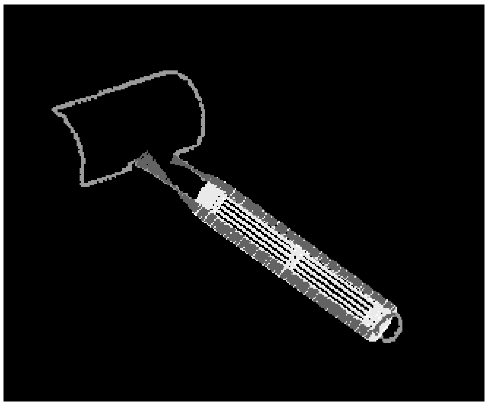
Superquad Parameter	**Part**	
	Head	**Handle**
a	37.19	37.19
a_1	0.45	0.22
a_2	0.45	0.22
a_3	0.69	1.14
t_x	-4.40	4.97
t_y	0.51	-3.88
t_z	-50.0	-50.0
r_{11}	0.49	0.54
r_{12}	-0.22	0.07
r_{13}	-0.84	0.84
r_{21}	-0.14	0.78
r_{22}	0.93	0.27
r_{23}	-0.33	-0.53
r_{31}	0.86	-0.26
r_{32}	0.28	0.96
r_{33}	0.42	0.09
ϵ_1	0.0	0.0
ϵ_2	1.0	1.0
bend_z	0.0	0.0
taper_z	0.0	0.0

Table 2: Recovered dimensions for the mallet.

Dimension	**Part**	
	Head	**Handle**
width	16.74	8.18
height	16.74	8.18
length	25.66	42.40

are free to be part of other recovered volumes. Since we have already recovered a stick, and its defining contours were not used to infer the blob, we can instantiate the handle in the image. The last step in recognizing the object is to satisfy the functional relation between the two parts which is mapped into a spatial constraint on the part junction. Since the computed relative orientation of the two parts is such that their z axes are orthogonal (> 60 deg in our qualitative partitioning of angles), and since the junction occurs at the end of the handle and at the middle of the head, the algorithm successfully verifies the hammer in the image.

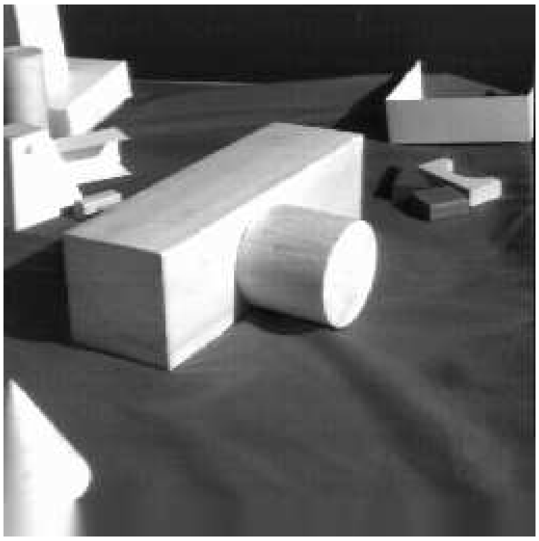




(a) (b)

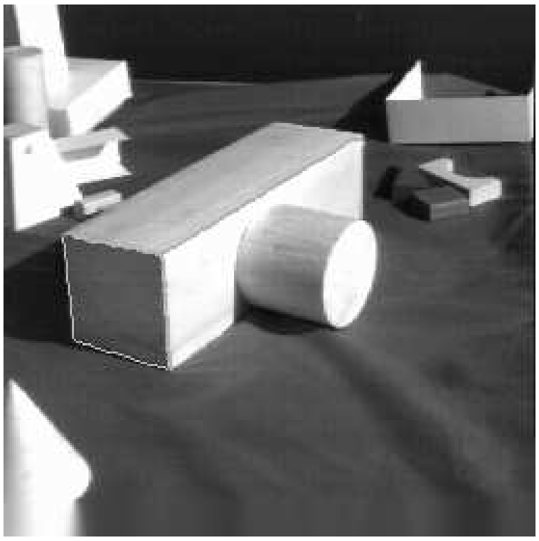
Figure 9: Recovered superquads: (a) mallet head; (b) mallet handle. Intermediate grey values along contour portions represent locations of image forces acting on the superquad. Due to the absence of forces at the junction between the two parts (no contours at the junction end of the handle), the fitted handle was not "pulled" all the way to the junction.

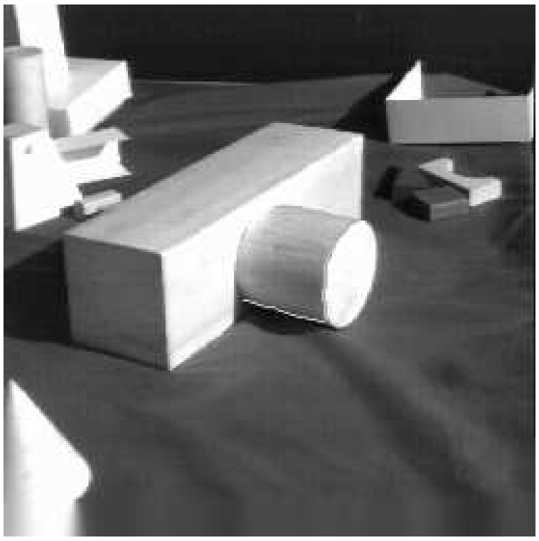
In the second example, we apply our function-based unexpected object recognition approach to a scene containing a short cylinder attached to the side of a block; the image is shown in Figure 10(a), and the segmented region image is shown in Figure 10(b). The most likely qualitative volumes recovered from the image are shown in Figures 11(a) and (b), respectively, while the fitted models are shown in Figures 12 (a) and (b), respectively.



(a) (b)

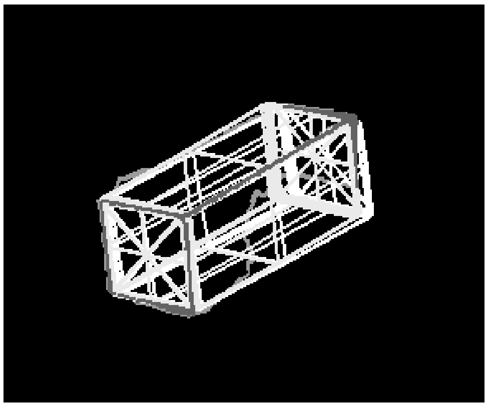
Figure 10: (a) Original image; (b) segmented region image.

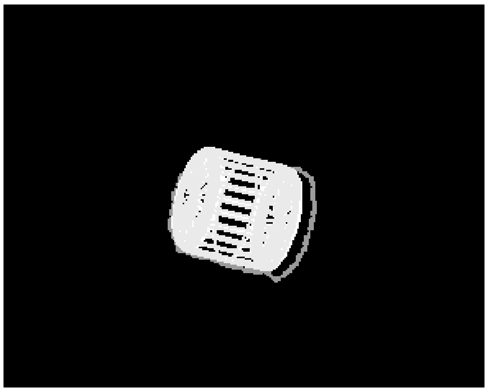




(a) (b)

Figure 11: Qualitative shape recovery: (a) most likely volume recovered for the block; (b) most likely volume recovered for the cylinder. Those contours used in inferring the recovered volume are highlighted in the image.





(a) (b)

Figure 12: Recovered superquads: (a) block; (b) cylinder. Intermediate grey values along contour portions represent locations of image forces acting on the superquad.

From the recovered superquad parameters in Table 1, we can proceed to classify each part as either a stick, a strip, a plate, or a blob according to equations 11, 12, and 13 in Section 2.1.1; the results are shown in Table 4. Using equations 1, 2, 3, and 4, and again defining two dimensions as similar if the ratio of the bigger to the smaller is less than 4:1 (the width:height:length ratios for the two parts are: 1:1:2.51 for the block and 1:1:0.89 for the cylinder), both the block and the cylinder are classified as blobs. Although their connection position and orientation is consistent with the hammer model, this model requires that the handle be a stick. The unknown object

Table 3: Recovered superquad parameters for the unknown object.

Superquad Parameter	**Part**	
	Block	**Cylinder**
a	37.19	37.19
a_1	0.54	0.45
a_2	0.54	0.45
a_3	1.36	0.40
t_x	-2.77	1.64
t_y	0.32	-0.06
t_z	-50.0	-50.0
r_{11}	0.78	0.39
r_{12}	-0.11	0.02
r_{13}	-0.62	0.92
r_{21}	-0.16	0.49
r_{22}	0.92	0.84
r_{23}	-0.36	-0.23
r_{31}	0.61	-0.78
r_{32}	0.38	0.54
r_{33}	0.70	0.32
ϵ_1	0.05	0.1
ϵ_2	0.05	1.0
bend$_z$	0.0	0.0
taper$_z$	0.0	0.0

Table 4: Recovered dimensions for the unknown object.

Dimension	**Part**	
	Block	**Cylinder**
width	20.08	16.74
height	20.08	16.74
length	50.58	14.88

cannot, therefore, be classified as a hammer.

6 Limitations

The domain of hand tools defines a simple, one-to-one mapping between an object's functional primitives and relations and the corresponding shape primitives and relations. In more general cases, the mapping from shape primitives to functional primitives may be many-to-one, and a more elaborate reasoning strategy is required to support the inference of a functional primitive from a collection of interacting shape primitives. Nevertheless, we believe that such a reasoning mechanism must operate at the level of an object's coarse volumetric parts.

The object representation described in this paper is appropriate for objects composed of simple volumetric parts. Furthermore, it supports only functionality that is defined in terms of an object's shape. Functions that are based on color, texture, or (more importantly) motion are not currently supported, but in our current work we are enhancing our representation to include the motions of an object's parts.

7 Conclusions

We have presented an approach to function-based object recognition that reasons about the functionalities of an object's parts. Previous approaches have relied on global object features, often ignoring the problem of object segmentation and thereby restricting themselves to range maps of unoccluded scenes. We extend the popular "recognition by parts" shape recognition framework to support "recognition by functional parts", by combining a set of functional primitives and their relations with a set of abstract volumetric shape primitives and their relations. We show how these shape primitives and relations can easily be recovered from superquadric ellipsoids which, in turn, can be recovered from either range or intensity images of occluded scenes. Furthermore, the proposed framework supports both unexpected (bottom-up) object recognition and expected (top-down) object recognition.

References

[1] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:11–23, 1981.

[2] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.

[3] R. Brooks. Model-based 3-D interpretations of 2-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:140–150, 1983.

[4] S. Dickinson. Part-based modeling and qualitative recognition. In A. Jain and P. Flynn, editors, *Three-Dimensional Object Recognition Systems*. Elsevier, Amsterdam, 1993.

[5] S. Dickinson and D. Metaxas. Using qualitative shape to constrain deformable model fitting. In *Proceedings, Sensor Fusion V, SPIE OE/Technology '92*, Boston, MA, November 1992.

[6] S. Dickinson, A. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3-D object recognition. *CVGIP: Image Understanding*, 55:130–154, 1992.

[7] S. Dickinson, A. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:174–198, 1992.

[8] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and superquadrics. In *Proceedings, Workshop on Interpretation of 3D Scenes*, pages 170–176, 1989.

[9] F. Ferrie, J. Lagarde, and P. Whaite. Recovery of volumetric descriptions from laser rangefinder images. In *Proceedings, ECCV '90*, pages 387–396, 1990.

[10] J. Gibson. *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston, 1979.

[11] A. Gupta. Surface and volumetric segmentation of 3D objects using parametric shape models. Technical Report MS-CIS-91-45, University of Pennsylvania, Philadelphia, PA, 1991.

[12] D. Huttenlocher. Three-dimensional recognition of solid objects from a two-dimensional image. Technical Report 1045, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1988.

[13] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5:195–212, 1990.

[14] D. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer, Norwell, MA, 1985.

[15] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:441–450, 1991.

[16] D. Marr. *Vision.* W. H. Freeman, San Francisco, CA, 1982.

[17] D. Metaxas and S. Dickinson. Integration of quantitative and qualitative techniques for deformable model fitting from orthographic, perspective, and stereo projections. In *Proceedings, Fourth International Conference on Computer Vision*, 1993.

[18] P. Mulgaonkar, L. Shapiro, and R. Haralick. Matching "sticks, plates and blobs" objects using geometric and relational constraints. *Image and Vision Computing*, 2:85–98, 1984.

[19] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[20] A. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4:107–126, 1990.

[21] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:715–729, 1991.

[22] N. Raja and A. Jain. Recognizing geons from superquadrics fitted to range data. *Image and Vision Computing*, 10:179–190, 1992.

[23] E. Rivlin, Y. Aloimonos, and A. Rosenfeld. Purposive recognition: A framework. Technical Report CAR-TR-2811, Center for Automation Research, University of Maryland, College Park, MD, 1991.

[24] F. Solina. Shape recovery and segmentation with deformable part models. Technical Report MS-CIS-87-111, University of Pennsylvania, Philadelphia, PA, 1987.

[25] L. Stark and K. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1097–1104, 1991.

[26] L. Stark and K. Bowyer. Generic recognition through qualitative reasoning about 3-D shape and object function. In *Proceedings, Computer Vision and Pattern Recognition*, pages 251–256, 1991.

[27] L. Stark and K. Bowyer. Indexing function-based categories for generic object recognition. In *Proceedings, Computer Vision and Pattern Recognition*, pages 795–797, 1992.

[28] L. Stark and K. Bowyer. Indexing function-based categories for generic object recognition. *CVGIP: Image Understanding*, to appear.

[29] L. Stark, L. Hall, and K. Bowyer. An investigation of methods of combining functional evidence for 3-D object recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, to appear.

[30] L. Stark, A. Hoover, D. Goldgof, and K. Bowyer. Function based recognition from incomplete knowledge of shape. In *Proceedings, Workshop on Qualitative Vision*, pages 11–22, 1993.

[31] M. Sutton, L. Stark, and K. Bowyer. Function-based generic recognition for multiple object categories. In A. Jain and P. Flynn, editors, *Three-Dimensional Object Recognition Systems.* Elsevier, Amsterdam, 1992.

[32] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:703–714, 1991.

[33] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 36:91–123, 1988.

[34] D. Thompson and J. Mundy. Model-directed object recognition on the connection machine. In *Proceedings, DARPA Image Understanding Workshop*, pages 93–106, 1987.

[35] P. Winston, T. Binford, B. Katz, and M. Lowry. Learning physical description from functional descriptions, examples, and precedents. In *Proceedings, AAAI*, pages 433–439, 1983.